Theoretical Computer Science 1003 (2024) 114618

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

A mechanism design approach for multi-party machine learning ☆

Mengjing Chen [a], Yang Liu [b], Weiran Shen [c,*], Yiheng Shen [d], Pingzhong Tang [a], Qiang Yang [e]

[a] Tsinghua University, Beijing, China
[b] WeBank Co., Ltd., Shenzhen, Guangdong Province, China
[c] Renmin University of China, Beijing, China
[d] Duke University, Durham, NC, USA
[e] Hong Kong University of Science and Technology, Hong Kong, China

ARTICLE INFO

Keywords:
Mechanism design
Federated learning
Incentive design

ABSTRACT

In a multi-party machine learning system, different parties cooperate on optimizing towards better models by sharing data in a privacy-preserving way. A major challenge in learning is the incentive issue. For example, if there is competition among the parties, one may strategically hide their data to prevent other parties from getting better models.

In this paper, we study the problem through the lens of mechanism design and incorporate the features of multi-party learning in our setting. First, each agent's valuation has externalities that depend on others' types and actions. Second, each agent can only misreport a type lower than his true type, but not the other way round. We provide the optimal truthful mechanism in the separable utility setting, as well as necessary and sufficient conditions for truthful mechanisms in general cases. Finally, we propose an algorithm to find the desirable mechanism that is truthful, individually rational, efficient and weakly budget-balanced, and analyze the computational complexity of the algorithm.

1. Introduction

In multi-party machine learning, a group of parties cooperates on optimizing towards better models. This concept has attracted much attention recently [1–3]. The advantage of this approach is that, it can make use of the distributed datasets and computational power to learn a powerful model that anyone in the group cannot achieve alone.

To make multi-party machine learning practical, a large body of studies focuses on preserving data privacy in the learning process [4,5,2]. However, the incentive issues in the multi-party learning have largely been ignored in most previous studies, which results in a significant reduction in the effectiveness when putting their techniques into practice. Previous work usually let all the parties share the same global model with the best quality regardless of their contributions. This allocation works well when there are no conflicts of interest among the parties. For example, an app developer wants to use the users' usage data to improve the user experience. All users are happy to contribute data since they can all benefit from such improvements [6].

☆ This article belongs to Section A: Algorithms, automata, complexity and games, Edited by Paul Spirakis.

* Corresponding author.
E-mail addresses: cmj16@tsinghua.org.cn (M. Chen), yangliu@webank.com (Y. Liu), shenweiran@ruc.edu.cn (W. Shen), ys341@duke.edu (Y. Shen), kenshinping@gmail.com (P. Tang), qyang@cse.ust.hk (Q. Yang).

https://doi.org/10.1016/j.tcs.2024.114618
Received 30 December 2022; Received in revised form 15 November 2023; Accepted 30 April 2024

Available online 7 May 2024
0304-3975/

When the parties are competing with one another, they may be unwilling to participate in the learning process since their competitors can also benefit from their contributions. Consider the case where companies from the same industry are trying to adopt federated learning to level up the industry's service qualities. Improving other companies' services can possibly harm one's own market share, especially when there are several monopolists that own most of the data.

Such a cooperative and competitive relation poses an interesting challenge that prevents the multi-party learning approach from being applied to a wider range of environments. In this paper, we view this problem from the multi-agent system perspective, and address the incentive issues mentioned above with the mechanism design theory.

Another key feature of our setting is that each agent cannot "make up" a dataset that is of a higher quality than their actual one. Thus the reported type of an agent is capped by their true type. The setting that agents can never over-report is common in practice. One straightforward example is sports competitions where athletes can show lower performance than their actual abilities but usually cannot over-perform. The restriction on the action space poses more constraints on agents' behaviors, and thus allows more flexibility in the design space.

We first formulate the problem mathematically, and then apply techniques from the mechanism design theory to analyze it. Our model is more general than the standard mechanism design framework, and is also able to describe other similar problems involving both cooperation and competition.

We make the following contributions in this paper:

- We model and formulate the mechanism design problem in multi-party machine learning, and identify the differences between our setting and the other mechanism design settings.
- For the case with separable utility functions, we provide the revenue-optimal and truthful mechanism. For the case with general valuation functions, we provide both necessary and the sufficient conditions for all truthful and individually rational mechanisms.
- We design an algorithm to find the mechanisms that guarantee individual rationality, truthfulness, efficiency and weak budget balance simultaneously when the valuation functions are given.

1.1. Related work

In our setting, agents have restricted action spaces, i.e., they can never report types exceeding their actual types. There is a series of works that focus on mechanism design with a restricted action space [7,8]. The discrete-bid ascending auctions [9,10,8] specify that all bidders' action spaces are the same bid level set. Previous works focus on mechanisms with independent values and discrete restricted action spaces, while we study the interdependent values and continuous restricted action spaces setting.

In multi-party machine learning, the learned model can be copied and distributed to as many agents as possible, so the supply is unlimited. A related literature focuses on designing mechanisms in unlimited supply settings such as selling digital goods [11–13]. However, the seller sells the same item to buyers while in our setting we can allocate models with different qualities to different agents.

Redko and Laclau [14] study the optimal strategies of agents for collaborative machine learning problems. Both their work and ours capture the cooperation and competition among the agents, but they only consider the case where agents reveal their total datasets to participate while agents can choose to contribute only a fraction in our setting. Kang et al. [15] study the incentive design problem for federated learning, but all their results are about a non-competitive environment, which may not hold in real-world applications.

Our work contributes to the growing body of literature on incentive mechanism design for federated learning [16,17]. Jia et al. [18], Song et al. [19] design mechanisms based on the Shapley value and Ding et al. [20] apply the contract theory. However, these existing works do not consider the interdependent values of the participants and type-dependent action space as in our model.

Our setting is also related to the so-called interdependent value setting [21–23] in the mechanism design literature. With the competition among the agents in our setting, the utilities of them depend on the final model they obtain from the platform, which, in turn, depend on their *reported types*. However, a key difference between our setting and the standard interdependent value setting is that, in the standard interdependent value setting, an agent's type affects other agents' values regardless of what they report.

2. Preliminaries

In this section, we introduce the general concepts of mechanism design and formulate the multi-party machine learning as a mechanism design problem. A multi-party learning consists of a central platform and several parties (called agents hereafter). The agents serve their customers with their models trained by the platform using a multi-party learning framework. Each agent can choose whether to participate in the framework, and if an agent does not participate, then they train their model with only their own data. The platform requires all the participating agents to contribute their data in a privacy-preserving way and trains a model for each participant using a (weighted) combination of all the contributions. Then the platform returns the trained models to the agents.

An important problem in this process is the incentive issue. For example, if the participants have conflicts of interest, then they may only want to make use of others' contributions but are not willing to contribute with all their own data. To align their incentives, we allow the platform to charge the participants according to some predefined rules.

In this paper, we analyze this problem from the angle of mechanism design, and aim to design allocation and payment rules that encourage all agents to join the multi-party learning as well as to contribute all their data.

2.1. Valid data size (type)

Denote by $N = \{1, 2, \ldots, n\}$ the set of n agents. Each agent has a private dataset and all their datasets are disjoint. In machine learning, a model's performance usually depends crucially on the size of the dataset used to train it. For ease of presentation, we assume that a model is fully characterized by its quality Q (e.g., the prediction accuracy), which only depends on the amount of data used to train it. We normalize the model quality Q such that $Q \in [0, 1]$, where $Q = 0$ represents a random guessing model and $Q = 1$ represents a perfect model. Let s be the size of a dataset and $Q(s)$ be the quality of the resulting model. We have:

$$Q(0) = 0 \quad \text{and} \quad Q(s) < 1, \forall s.$$

In a multi-party machine learning process, it is possible that an agent only contributes part of their dataset, or even strategically add noisy or fake data to maximize their own utility. In this case, the model quality may not depend on the size of the dataset. To tackle this problem, we measure the contribution of a dataset to the corresponding model by its *valid data size*, and assume that the model quality increases with respect to the valid data size.

Assumption 1. *The model quality Q is bounded and monotone increasing with respect to the valid data size $s \geq 0$ of the training data:*

$$Q(s') > Q(s), \forall s' > s.$$

Let $t_i \in \mathbb{R}_+$ be the valid data size of agent i's private dataset. We call t_i the agent's *type*. Clearly, by using only a subset of the private dataset or adding fake data, an agent can only lower their data quality, and thus decreases the contribution to the trained model. As a result, we assume that each agent cannot contribute to the platform with a dataset higher than their true type:

Assumption 2. *Each agent i can only report a type lower than his true type t_i, i.e., the action space of agent i is $[0, t_i]$.*

2.2. Learning protocol

In this section, we describe the learning protocol that could enable the implementation of our mechanism.

In mechanism design theory, it is crucial for the mechanism to obtain information about the types of the participating agents, as such information would ultimately affect the final outcome of the mechanism. In standard mechanism design settings, mechanism designers can usually require the participating agents to report their types directly to the mechanism. However, in the multi-party learning process, the model is trained in a privacy-preserving way and the agents do not reveal their datasets to the platform. In order for the platform to know the agents' valid data size t_i, we further assume that the platform has a validation dataset, which can be used to compute the quality of any model.

With such a validation dataset, a naïve idea is to ask all the agents to submit the best model that they can possibly obtain by using their own dataset in the beginning. Then the platform computes the model quality q_i using the validation dataset and gets the agent's valid data size t_i by $t_i = Q^{-1}(q_i)$. The agent type t_i can be used by the platform in the later training process.

However, this naïve idea can be problematic, since it is possible that an agent reports t_i in the beginning but only contributes $t_i' < t_i$ in the actual training process. In the extreme case where all agents contribute nothing to the training process, the platform will fail to train a useful model. To address this issue, the platform can train n additional models simultaneously, with the i-th model trained only using the data from agent i. During the training process, the platform can apply secure multi-party computation techniques, such as homomorphic encryption [5,24], to prevent the agent from knowing which model is sent to him to compute the update. And after the training, the platform can compute the quality t_i' of the i-th model again using the validation dataset. If the qualities t_i' and t_i match, we know with high probability that the dataset contributed by the agent is consistent with the type he reports. Otherwise, the platform can just exclude the agent and start over the training process again.

The above protocol only ensures that the type reported by each agent is the same as the type he uses in the actual training process with. To encourage all agents to join and contribute all their data, we still need to design mechanisms with desirable properties, to which we devote the rest of the paper.

2.3. Mechanism

Let $t = (t_1, t_2, \ldots, t_n)$ and $t_{-i} = (t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n)$ be the type profile of all agents and all agents without i, respectively. Given the reported types of agents, a mechanism specifies a numerical allocation and payment for each agent, where the allocation is a model of certain quality trained in the multi-party learning process. Formally, we have:

Definition 1 *(Mechanism).* A mechanism $\mathcal{M} = (x, p)$ is a tuple, where

- $x = (x_1, x_2, \cdots, x_n)$, where x_i: $\mathbb{R}_+^n \mapsto \mathbb{R}$ is the allocation function for agent i, which takes the agents' reported types as input and decides the model quality for agent i as output;
- $p = (p_1, p_2, \cdots, p_n)$, where p_i: $\mathbb{R}_+^n \mapsto \mathbb{R}$ is the payment function for agent i, which takes the agents' reported types as input and specifies how much agent i should pay to the mechanism.

Let t' denote the reported type profile of all agents. Here, the allocation function has a natural upper bound: $x_i(t') \le Q(\sum_i t'_i)$, i.e., the best model the platform can allocate is the one trained with all the data submitted by the agent. Therefore, any feasible allocation must satisfy $x_i \in [0, Q(\sum_i t'_i)]$. In a competitive environment, a strategic agent may not use the model they receive from the platform but train a private model with their own data instead. Thus the final model quality depends on both the allocation and the actual type of an agent. We use valuation function $v_i(x(t'), t)$ to measure the profit of agent i.

Definition 2 *(Valuation)*. We consider valuation functions $v_i(x(t'), t)$ that depend not only on the allocation outcome $x(t')$ where t' is the reported type profile, but also on the actual type profile t.

We assume the model agent i uses to serve their customers is:

$$q_i = \max\{x_i(t'), Q(t_i)\},$$

where $Q(t_i)$ is the model trained with his own data. The valuation of agent i depends on the final model qualities of all agents due to their competition. Hence v_i can also be expressed as $v_i(q_1, \dots, q_n)$.

We make the following assumption on agent i's valuation:

Assumption 3. *Agent i's valuation is monotone increasing with respect to true type t_i when the outcome x is fixed.*

$$v_i(x, t_i, t_{-i}) \ge v_i(x, \hat{t}_i, t_{-i}), \forall x, \forall t_i \ge \hat{t}_i, \forall t_{-i}, \forall i.$$

This is because possessing more valid data will not lower one's valuation. Otherwise, an agent is always able to discard part of his dataset to make his true type t'_i. Suppose that each agent i's utility $u_i(t, t')$ has the form:

$$u_i(t, t') = v_i(x(t'), t) - p_i(t'),$$

where t and t' are true types and reported types of all agents respectively. As we mentioned above, an agent may lie about his type in order to benefit from the mechanism. The mechanism should incentivize truthful reports to keep agents from lying.

Definition 3 *(Ex-post Incentive Compatibility (ex-post IC))*. A mechanism is said to be ex-post incentive compatible, or ex-post truthful, if reporting truthfully is always the best response for each agent when the other agents report truthfully:

$$u_i(x(t_i, t_{-i}), t) \ge u_i(x_i(t'_i, t_{-i}), t), \forall t_i \ge t'_i, \forall t_{-i}, \forall i.$$

For ease of presentation, we say agent i reports $\emptyset$ if he chooses not to participate (so we have $x_i(\emptyset, t_{-i}) = 0$ and $p_i(\emptyset, t_{-i}) = 0$). Note that not participating is different from participating but reporting nothing ($t'_i = 0$). A non-participating agent can only use the model trained with his own data, while a non-contributing participant can still possibly get from the platform a model that is better than his own model. Therefore, with such a distinction, an agent with no data can still buy data (through the model from the platform) from other agents.

To encourage the agents to participate in the mechanism, the following property should be satisfied:

Definition 4 *(Individual Rationality (IR))*. A mechanism is said to be individually rational, if no agent loses by truthful participation when the other agents report truthfully:

$$u_i(x(t_i, t_{-i}), t) \ge u_i(x(\emptyset, t_{-i}), t), \forall t_i, t_{-i}, \forall i.$$

The revenue and welfare of a mechanism are defined to be all the payments collected from the agents and all the valuations of the agents.

Definition 5. The revenue and welfare of a mechanism (x, p) are:

$$\textsc{Rev}(x, p) = \sum_{i=1}^{n} p_i(t'),$$

$$\textsc{Wel}(x, p) = \sum_{i=1}^{n} v_i(x, t).$$

We say that a mechanism is *efficient* if

$$(x, p) = \arg\max_{(x,p)} \textsc{Wel}(x, p).$$

A mechanism is *weakly budget-balanced* if it never loses money.

Definition 6 *(Weak Budget Balance).* A mechanism is weakly budget-balanced if:

$$\textsc{Rev}(x, p) \geq 0, \forall t.$$

Definition 7 *(Desirable Mechanism).* We say a mechanism is desirable if it is ex-post IC, IR, efficient and weakly budget-balanced.

A key characteristic of our setting is that each agent cannot submit a dataset that is of a higher quality than their true dataset. This means an agent with type t_i cannot over-report type t_i' with $t_i' > t_i$, i.e., the action space of each agent is $[0, t_i]$ and depends on their true type t_i. This is different from the standard mechanism design setting, where both over-reporting and under-report are allowed. Our setting puts a stronger restriction on agents' behaviors, thus has a larger design space.

Another difference from the standard mechanism design setting is that, in our setting, the agents do not have the "exit choice" (not participating in the mechanism and getting 0 utility). This is due to the motivation of this paper: companies from the same industry are trying to improve their service quality, and they are always in the game regardless of their choices. A non-participating company may even have a negative utility because of loss of market share if all other companies improved their services.

3. Separable externality setting

In our setting, each agent's utility may also depend on the models that other agents actually use. Such externalities lead to interesting and complicated interactions between the agents. For example, by contributing more data, one may improve the others' model quality, and end up harming their own market share. In this section, we study the setting where agents have *separable* externalities.

Definition 8 *(Separable Valuation).* Let q_i be the final selected model quality of the agent and q_{-i} be the profile of model qualities of all the agents except i. A valuation function is separable if it is in the form:

$$v_i(q_i, q_{-i}) = F_i(q_i) + \theta_i(q_{-i}),$$

where F_i is monotone increasing and θ_i is an arbitrary function.

Example 1. Let's consider a special separable valuation: the linear externality setting, where the valuation for each agent is defined as $v_i = \sum_j \alpha_{ij} q_j$ with q_j being the model that agent j uses. The externality coefficient α_{ij} means the influence of agent j to agent i and captures either the competitive or cooperative relations among agents. If the increase of agent j's model quality imposes a negative (positive) effect on agent i's utility (e.g. major opponents or collaborators in the market), α_{ij} would be negative (positive). Additionally, α_{ii} should always be positive, naturally.

In the linear externality setting, the efficient allocation is straightforward. For each agent i, we give i the training model with the best possible quality if $\sum_j \alpha_{ji} \geq 0$. Otherwise, no model is given to agent i.

We introduce a payment function called *maximal exploitation payment*, and show that the mechanism with efficient allocation and the maximal exploitation payment guarantees IR, ex-post IC, efficiency and revenue optimum.

Definition 9 *(Maximal Exploitation Payment (MEP)).* For a given allocation function x, if the agent i reports a type t_i' and the other agents report t_{-i}', the maximal exploitation payment is to charge agent i

$$p_i(t_i', t_{-i}') = v_i(x(t_i', t_{-i}'), t_i', t_{-i}') - v_i(x(\emptyset, t_{-i}'), t_i', t_{-i}').$$

We emphasize that our MEP mechanism and the VCG are quite different. The VCG charges each agent for the harm he causes to others due to his participation while the MEP charges each agent the profit he gets from the mechanism due to his participation. We will show that the MEP is truthful in the separable valuation setting in the following theorem, while it is already known that VCG cannot guarantee truthfulness in similar settings [25] (e.g., the interdependent setting).

Theorem 1. *Under the separable valuation setting, given any allocation function x, the mechanism $(x, \bar{p})$ where $\bar{p}$ is the MEP is IR, ex-post IC, and has the maximal revenue among all mechanisms (x, p) that are IR.*

Proof. Intuitively, the MEP rule charges agent i the profit he gets from an model that the mechanism allocates to him. If the mechanism charges higher than the MEP, an agent would have negative utility after taking part in. The IR constraint would then be violated. So it's easy to see that the MEP is the maximal payment among all IR mechanisms.

Then we prove that this payment rule also guarantees the ex-post IC property. It suffices to show that if an agent hides some data, no matter which model he chooses to use, he would never get more utility than that of truthful reporting. Let t_i and t_i' be the true and reported type of agent i.

Suppose that the agent i truthfully reports his type, i.e., $t_i' = t_i$. Since the payment function is defined to charge this agent until he reaches the valuation when he does not take part in the mechanism, the utility of this honest agent would be

$$u_i^0(t') = u_i^0(t_i, t'_{-i}) = F_i(Q(t_i)) + \theta_i(q_{-i}(\emptyset, t'_{-i})),$$

where the first equality is due to $t'_i = t_i$.

If the agent misreports his type, i.e., $t'_i \le t_i$. According to the MEP, the payment function for agent i would be

$$p_i(t'_i, t'_{-i}) = F_i(q_i(t'_i, t'_{-i})) + \theta_i(q_{-i}(t'_i, t'_{-i})) - F_i(Q(t'_i)) - \theta_i(q_{-i}(\emptyset, t'_{-i})).$$

It can be seen that the mechanism would never give an agent a worse model than the model trained by its reported data, otherwise the agents would surely select their private data to train models. Hence it is without loss of generality to assume that the allocation $x_i(t'_i, t'_{-i}) \ge Q(t'_i)$, $\forall t'_i, t'_{-i}, \forall i$. Thus we have $q_{-i}(t'_i, t'_{-i}) = x_{-i}(t'_i, t'_{-i})$.

We consider the following two cases.

Case 1: the agent chooses the allocation x_i. Since agent i selects the allocated model, we have $q_i = x_i(t'_i, t'_{-i})$. Then the utility of agent i would be

$$\begin{aligned}
u_i^1 =& v_i(t'_i, t'_{-i}) - p_i(t'_i, t'_{-i})\\
=& F_i(x_i(t'_i, t'_{-i})) + \theta_i(x_{-i}(t'_i, t'_{-i})) + F_i(Q(t'_i))\\
&+ \theta_i(x_{-i}(\emptyset, t'_{-i})) - F_i(x_i(t'_i, t'_{-i})) - \theta_i(x_{-i}(t'_i, t'_{-i}))\\
=& F_i(Q(t'_i)) + \theta_i(x_{-i}(\emptyset, t'_{-i})).
\end{aligned}$$

Because both F_i and Q are monotone increasing functions and $t_i \ge t'_i$, we have

$$u_i^1 \le F_i(Q(t_i)) + \theta_i(x_{-i}(\emptyset, t'_{-i})) = u_i^0.$$

Case 2: the agent chooses $Q(t_i)$. Since agent i selects the model trained by his private data, we have $q_i = Q(t_i)$. The final utility of agent i would be

$$\begin{aligned}
u_i^2 =& v_i(t'_i, t'_{-i}) - p_i(t'_i, t'_{-i})\\
=& F_i(Q(t_i)) + \theta_i(x_{-i}(t'_i, t'_{-i})) + F_i(Q(t'_i))\\
&+ \theta_i(x_{-i}(\emptyset, t'_{-i})) - F_i(x_i(t'_i, t'_{-i})) - \theta_i(x_{-i}(t'_i, t'_{-i}))\\
=& F_i(Q(t_i)) + F_i(Q(t'_i)) + \theta_i(x_{-i}(\emptyset, t'_{-i})) - F_i(x_i(t'_i, t'_{-i})).
\end{aligned}$$

Subtract the original utility from the both sides, then we have

$$\begin{aligned}
u_i^2 - u_i^0 =& F_i(Q(t_i)) + F_i(Q(t'_i)) + \theta_i(x_{-i}(\emptyset, t'_{-i}))\\
&- F_i(x_i(t'_i, t'_{-i})) - F_i(Q(t_i)) - \theta_i(x_{-i}(\emptyset, t'_{-i}))\\
=& F_i(Q(t'_i)) - F_i(x_i(t'_i, t'_{-i})).
\end{aligned}$$

Because $x_i(t'_i, t'_{-i}) \ge Q(t'_i)$, $\forall t'_i, t'_{-i}, \forall i$ and because F_i is a monotonically increasing function, we can get $u_i^2 - u_i^0 \le 0$. Therefore $\max\{u_i^1, u_i^2\} \le u_i^0$, lying would not bring more benefits to any agent, and the mechanism is ex-post IC. □

Corollary 1. *Any efficient allocation mechanism with MEP under the linear externality setting with all the linear coefficients $\alpha_{ji} \ge 0$ should be IR, ex-post IC, weakly budget-balanced and efficient.*

Proof. In Theorem 1 we know that the MEP mechanism is IR and ex-post IC. Since the linear coefficients are all positive and the externality setting is linear, any efficient mechanism would allocate the best model to all the agents. Since each agent gets a model with no less quality than his reported one and the payment is equal to the value difference between the case an agent truthfully report and the case he exits the mechanism. The agent's value is always larger than the value when he exits the mechanism. Then the payment is always positive and the mechanism should satisfy all of the four properties. □

4. General externality setting

In this section, we consider the general externality setting where the valuations of agents can have any forms of externalities. The restrictions on the action space and the value functions make it difficult to characterize ex-post IC and IR mechanisms. It is possible that given an allocation rule, there exist more than one mechanism with different payments that satisfy both ex-post IC and IR constraints. To understand what makes a mechanism ex-post IC and IR, we analyze some properties of truthful mechanisms in this section. For ease of presentation, we assume that the functions $v(\cdot)$, $x(\cdot)$ and $p(\cdot)$ are differentiable. However, even when the assumption does hold, our results still apply. In this case, the derivatives and partial derivative of non-differentiable or even discontinuous functions are called *general functions* or *distributions* (e.g., the Dirac delta function). In this paper, we do not discuss this in detail, but refer interested readers to [26, Chapter 6].

Theorem 2 *(Necessary Condition). If a mechanism (x, p) is both IR and ex-post IC, for all possible valuation functions satisfying Assumption 3, then the payment function satisfies $\forall t_i \geq t_i', \forall t_i, \forall t_{-i}, \forall i$,*

$$p_i(0, t_{-i}) \leq v_i(x(0, t_{-i}), 0, t_{-i}) - v_i(x(\emptyset, t_{-i}), 0, t_{-i}), \tag{1}$$

$$p_i(t_i, t_{-i}) - p_i(t_i', t_{-i}) \leq \int_{t_i'}^{t_i} \left. \frac{\partial v_i(x(s', t_{-i}), s, t_{-i})}{\partial s'} \right|_{s=s'} \mathrm{d}s', \tag{2}$$

where we view $v_i(x(t_i', t_{-i}), t_i, t_{-i})$ as a function of t_i, t_i' and t_{-i} for simplicity.

Proof. We first prove that Equation (1) holds. To guarantee the IR property, we need to ensure that for any t_i and t_{-i}, the following holds:

$$u_i(x(t_i, t_{-i}), t) \geq u_i(x(\emptyset, t_{-i}), t).$$

In particular, when $t_i = 0$, we have:

$$u_i(x(0, t_{-i}), 0, t_{-i}) \geq u_i(x(\emptyset, t_{-i}), 0, t_{-i}),$$

or equivalently,

$$v_i(x(0, t_{-i}), 0, t_{-i}) - p_i(0, t_{-i}) \geq v_i(x(\emptyset, t_{-i}), 0, t_{-i}) - 0,$$

which directly implies Equation (1).

To show Equation (2) must hold, we observe that

$$\begin{aligned}
&u_i(x(t_i, t_{-i}'), t_i, t_{-i}) - u_i(x(t_i', t_{-i}'), t_i', t_{-i}) \\
=&[v_i(x(t_i, t_{-i}'), t_i, t_{-i}) - p_i(t_i, t_{-i}')] - [v_i(x(t_i', t_{-i}'), t_i', t_{-i}) - p_i(t_i', t_{-i}')] \\
\geq&[v_i(x(t_i, t_{-i}'), t_i, t_{-i}) - p_i(t_i, t_{-i}')] - [v_i(x(t_i', t_{-i}'), t_i, t_{-i}) - p_i(t_i', t_{-i}')] \\
=&u_i(x(t_i, t_{-i}'), t_i, t_{-i}) - u_i(x(t_i', t_{-i}'), t_i, t_{-i}) \\
\geq&0,
\end{aligned} \tag{3}$$

where the first inequality is due to Assumption 3, and the last inequality is because of the ex-post IC property.

Rewriting Equation (3), we have:

$$\begin{aligned}
p_i(t_i, t_{-i}') - p_i(t_i', t_{-i}') &\leq v_i(x(t_i, t_{-i}'), t_i, t_{-i}) - v_i(x(t_i', t_{-i}'), t_i', t_{-i}) \\
&= \int_{t_i'}^{t_i} \frac{\mathrm{d}v_i(x(s', t_{-i}'), s(s'), t_{-i})}{\mathrm{d}s'} \mathrm{d}s'.
\end{aligned} \tag{4}$$

Fixing t_{-i} and t_{-i}', the total derivative of $v_i(x(s', t_{-i}'), s, t_{-i})$ is:

$$\begin{aligned}
&\mathrm{d}v_i(x(s', t_{-i}'), s, t_{-i}) \\
=&\frac{\partial v_i(x(s', t_{-i}'), s, t_{-i})}{\partial s'} \mathrm{d}s' + \frac{\partial v_i(x(s', t_{-i}'), s, t_{-i})}{\partial s} \mathrm{d}s.
\end{aligned}$$

View s as a function of s' and let $s(s') = s'$:

$$\begin{aligned}
&\frac{\mathrm{d}v_i(x(s', t_{-i}'), s(s'), t_{-i})}{\mathrm{d}s'} \\
=&\left. \frac{\partial v_i(x(s', t_{-i}'), s, t_{-i})}{\partial s'} \right|_{s=s'} + \frac{\partial v_i(x(s', t_{-i}'), s(s'), t_{-i})}{\partial s(s')} \frac{\mathrm{d}s(s')}{\mathrm{d}s'}.
\end{aligned}$$

Plug into Equation (4), and we obtain:

$$\begin{aligned}
&p_i(t_i, t_{-i}') - p_i(t_i', t_{-i}') \\
\leq&\int_{t_i'}^{t_i} \left. \frac{\partial v_i(x(s', t_{-i}'), s, t_{-i})}{\partial s'} \right|_{s=s'} + \int_{t_i'}^{t_i} \frac{\partial v_i(x(s', t_{-i}'), s(s'), t_{-i})}{\partial s(s')} \mathrm{d}s'.
\end{aligned}$$

Since the above inequality holds for any valuation function with $v_i(x, t_i, t_{-i}) \geq v_i(x, t_i', t_{-i}), \forall x, \forall t_{-i}, \forall t_i \geq t_i'$, we have:

$$p_i(t_i, t'_{-i}) - p_i(t'_i, t'_{-i}) \leq \int_{t'_i}^{t_i} \left.\frac{\partial v_i(x(s', t'_{-i}), s, t_{-i})}{\partial s'}\right|_{s=s'} \mathrm{d}s'. \qquad \square$$

Theorem 2 describes what the payment p is like in all ex-post IC and IR mechanisms. In fact, the conditions in Theorem 2 are also crucial in making a mechanism truthful. However, to ensure ex-post IC and IR, we still need to impose restrictions on the allocation rule.

Theorem 3 *(Sufficient Condition). Given any valuation function satisfying Assumption 3, a mechanism (x, p) satisfies both IR and ex-post IC, if for each agent i, for all $t_i \geq t'_i$, and all t_{-i}, Equations (1) and the following two hold*

$$t'_i \in \arg\min_{t_i : t_i \geq t'_i} \frac{\partial v_i(x(t'_i, t_{-i}), t_i, t_{-i})}{\partial t'_i} \tag{5}$$

$$\begin{aligned} &p_i(t_i, t_{-i}) - p_i(t'_i, t_{-i}) \\ &\leq \int_{t'_i}^{t_i} \left.\frac{\partial v_i(x(s', t_{-i}), s, t_{-i})}{\partial s'}\right|_{s=s'} \mathrm{d}s' - \int_{t'_i}^{t_i} \frac{\partial v_i(x(\emptyset, t_{-i}), s, t_{-i})}{\partial s} \mathrm{d}s. \end{aligned} \tag{6}$$

Proof. Equation (5) indicates that the function $\frac{\partial v_i(x(t'_i, t'_{-i}), t_i, t_{-i})}{\partial t'_i}$ is minimized at t'_i:

$$\left.\frac{\partial v_i(x(t'_i, t'_{-i}), s, t_{-i})}{\partial t'_i}\right|_{s=t'_i} \leq \frac{\partial v_i(x(t'_i, t'_{-i}), t_i, t_{-i})}{\partial t'_i}. \tag{7}$$

Therefore, we have

$$\begin{aligned} &u_i(x(t_i, t'_{-i}), t_i, t_{-i}) - u_i(x(t'_i, t'_{-i}), t_i, t_{-i}) \\ &= \int_{t'_i}^{t_i} \frac{\partial v_i(x(s', t'_{-i}), t_i, t_{-i})}{\partial s'} \mathrm{d}s' - p_i(t_i, t'_{-i}) + p_i(t'_i, t'_{-i}) \\ &\geq \int_{t'_i}^{t_i} \left.\frac{\partial v_i(x(s', t'_{-i}), s, t_{-i})}{\partial s'}\right|_{s=s'} \mathrm{d}s' - p_i(t_i, t'_{-i}) + p_i(t'_i, t'_{-i}) \\ &\geq \int_{t'_i}^{t_i} \frac{\partial v_i(x(\emptyset, t'_{-i}), s, t_{-i})}{\partial s} \mathrm{d}s, \end{aligned} \tag{8}$$

where the two inequalities are due to Equation (7) and (6), respectively. Since $v_i(x, t_i, t_{-i}) \geq v_i(x, t'_i, t_{-i})$, $\forall x$, $\forall t_{-i}$, $\forall t_i \geq t'_i$ indicates $\frac{\partial v_i(x(\emptyset, t'_{-i}), s, t_{-i})}{\partial s} \geq 0$, the above inequality shows that the mechanism guarantees the ex-post IC property.

To prove that the mechanism is IR, we first observe that

$$\begin{aligned} &[u_i(x(t_i, t'_{-i}), t_i, t_{-i}) - v_i(x(\emptyset, t'_{-i}), t_i, t_{-i})] - [u_i(x(t'_i, t'_{-i}), t'_i, t_{-i}) \\ &- v_i(\emptyset, x(t_{-i}), t'_i, t_{-i})] \\ &= u_i(x(t_i, t'_{-i}), t_i, t_{-i}) - u_i(x(t'_i, t'_{-i}), t'_i, t_{-i}) - \int_{t'_i}^{t_i} \frac{\partial v_i(x(\emptyset, t'_{-i}), s, t_{-i})}{\partial s} \mathrm{d}s \\ &\geq u_i(x(t_i, t'_{-i}), t_i, t_{-i}) - u_i(x(t'_i, t'_{-i}), t_i, t_{-i}) - \int_{t'_i}^{t_i} \frac{\partial v_i(x(\emptyset, t'_{-i}), s, t_{-i})}{\partial s} \mathrm{d}s \\ &\geq 0, \end{aligned}$$

where the two inequalities are Assumption 3 and Equation (8). Letting $t'_i = 0$ using Equation (2), we get:

$$u_i(x(t_i, t'_{-i}), t_i, t_{-i}) - v_i(x(\emptyset, t'_{-i}), t_i, t_{-i})$$

$$\geq u_i(x(0,t'_{-i}),0,t_{-i}) - v_i(x(\emptyset,t'_{-i}),0,t_{-i})$$
$$= v_i(x(0,t'_{-i}),0,t_{-i}) - p_i(0,t'_{-i}) - v_i(x(\emptyset,t'_{-i}),0,t_{-i})$$
$$\geq 0. \quad \square$$

5. Algorithm for finding a desirable mechanism

In the linear externality setting, we provide a mechanism that satisfies all the desirable properties. But this mechanism is not applicable to all valuation functions in the general setting, since the existence of a desirable mechanism depends on the agents' actual valuation functions. We provide an algorithm, that given the agents' valuations, computes whether such a mechanism exists, and outputs the one that optimizes revenue, if any.

Since each agent can only under-report, according to the IR property, we must have:

$$u_i(x(t_i,t_{-i}),t) \geq u_i(x(\emptyset,t_{-i}),t), \forall t, \forall i.$$

Equivalently, we get $\forall t, \forall i$,

$$u_i(x(\emptyset,t_{-i}),t) \leq v_i(x(t_i,t_{-i}),t) - p_i(t_i,t_{-i}),$$
$$p_i(t_i,t_{-i}) \leq v_i(x(t_i,t_{-i}),t) - u_i(x(\emptyset,t_{-i}),t),$$
$$p_i(t) \leq v_i(x(t_i,t_{-i}),t) - u_i(x(\emptyset,t_{-i}),t).$$

For simplicity, we define the upper bound of $p(t')$ as

$$\overline{p(t)} \triangleq \{ v_i(x(t_i,t_{-i}),t) - u_i(x(\emptyset,t_{-i}),t).$$

The ex-post IC property requires that $\forall t_i \geq t'_i, \forall t_{-i}, \forall i$,

$$u_i(x(t_i,t_{-i}),t) \geq u_i(x(t'_i,t_{-i}),t).$$

A little re-arrangement gives:

$$p_i(t_i,t_{-i}) - p_i(t'_i,t_{-i}) \leq v_i(x(t_i,t_{-i}),t) - v_i(x(t'_i,t_{-i}),t) \triangleq Gap_i(t'_i,t_i,t_{-i}).$$

Note that the inequality correlations between the payments form a system of difference constraints. The form of update of the payments is almost identical to that of the shortest path problem. In fact, the connection between pricing problems and shortest path problems has long been observed in the literature. For example, the problem of finding the clearing price of a market with multiple buyers and sellers can be formulated as a maximum or a minimum flow problem, which can be solved by the successive shortest path algorithm (which is equivalent to the Hungarian algorithm [27,28] under certain conditions). Another line of work considers using auction algorithms [29] to solve shortest path problems [30–32].

We make use of such a connection to design our algorithm. We assume that all the value functions are common knowledge, the efficient allocation is then determined because the mechanism always chooses the one that maximizes the social welfare. Thus it suffices to figure out whether there is a payment rule $p(t')$ which makes the mechanism IR, ex-post IC and weakly budget-balanced. Since the valid data size for each agent is bounded in practice, we assume the mechanism only decides the payment functions on the data range $[0, D]$, and discretize the type space into intervals of length ϵ, which is also the minimal size of the data. Thus each agent's type is a multiple of ϵ. Note that since the utility function is general, all the points in the action space would influence the properties and existence of the mechanism, thus it is necessary to enumerate all the points in the space, which already takes exponential time. Therefore, the running time of any algorithm that finds a desirable mechanism in the general setting is exponential in n.

Nevertheless, we give Algorithm 1 that checks if a desirable mechanism exists. The algorithm also outputs a desirable mechanism that is revenue maximizing if such mechanisms exist.

We now show the correctness of Algorithm 1.

Theorem 4. *Taking agents' valuation functions as input, Algorithm 1 outputs the answer of the decision problem of whether there exists a mechanism that guarantees IR, ex-post IC, efficiency and weak budget balance simultaneously, and specifies the payments that achieve maximal revenue if the answer is yes.*

Proof. Suppose that there is a larger payment for agent i such that $p_i(t') > p_i^{\max}(t')$ where t' is the profile of reported types. In the process of our algorithm, the $p_i^{\max}(t')$ is the minimal path length from VB_{-i} to $V_{t_i}t_{-i}$, denoted by $(VB_{-i}, V_{t_{i1}}t_{-i}, V_{t_{i2}}t_{-i}, \cdots, V_{t_{ik}=t'_i}t_{-i})$. By the definition of edge weight, we have the following inequalities:

$$p_i(t_{i1},t_{-i}) \leq \overline{p_i(t_{i1},t_{-i})},$$
$$p_i(t_{i2},t_{-i}) - p_i(t_{i1},t_{-i}) \leq Gap_i(t_{i1},t_{i2},t_{-i}),$$
$$\vdots$$

Algorithm 1: Finding desirable mechanisms

input: Agents' valuation functions v.
Use the function v_i to calculate all the $Gap_i(t_i', t_i, t_{-i})$ and $\overline{p_i(t_i, t_{-i})}$ for each i;
Initialize all $p_i^{max}(t_i, t_{-i})$ to be $\overline{p_i(t_i, t_{-i})}$ for each i;
for $i = 1$ *to* n **do**
  for $t_{-i} = (\emptyset, \emptyset, \cdots, \emptyset)$ *to* $(D, D, \cdots, D)$ *(increment = ϵ on each dimension)* **do**
    Build an empty graph;
    For each $p_i(t_i, t_{-i})$, construct a vertex $V_{t_i t_{-i}}$ and insert it into the graph;
    Construct a base vertex $VB_{t_{-i}}$ which denotes the payment zero into the graph;
    for $t_i = 0$ *to* D *(increment = ϵ)* **do**
      Add an edge from $VB_{t_{-i}}$ to $V_{t_i t_{-i}}$ with weight $\overline{p(t_i, t_{-i})}$;
      for $t_i' = 0$ *to* t_i *(increment = ϵ)* **do**
        Add an edge with weight $Gap_i(t_i', t_i, t_{-i})$ from $V_{t_i' t_{-i}}$ to $V_{t_i t_{-i}}$;
    Use the Single-Source Shortest-Path algorithm to find the shortest path from $VB_{t_{-i}}$ to all the other vertices. These are the maximum solutions $p_i^{max}(t_i, t_{-i})$ for each payment case;
    if $\sum_{j=1}^{n} p_j^{max}(t) < 0$ **then**
      return There is no desirable mechanism.
return p_i^{max} as the payment functions.

$$p_i(t_{ik}, t_{-i}) - p_i(t_{i(k-1)}, t_{-i}) \le Gap_i(t_{i1}, t_{i2}, t_{-i}).$$

Adding these inequalities together, we get

$$p_i(t') \le \overline{p_i(t_{i1}, t_{-i})} + \sum_{j=1}^{k-1} Gap_i(t_{ij}, t_{i(j+1)}, t_{-i}) = p_i^{\max}(t').$$

If $p_i(t') < p_i^{\max}(t')$ holds, this would violate at least 1 of the k inequalities above. If the first inequality is violated, the mechanism would not be IR, by the definition of $\overline{p_i(t_{i1}, t_{-i})}$. If any other inequality is violated, the mechanism would not be ex-post IC, by the definition of $Gap_i(t_{ij}, t_{i(j+1)}, t_{-i})$.

On the other hand, if we select $p_i^{\max}(t')$ to be payment of agent i, all the inequalities should be satisfied, otherwise the shortest path would be updated to a smaller length.

Therefore the $p_i^{\max}(t')$ must be the maximum payment for agent i. If the maximal payment sum up to less than 0, there would obviously be no mechanism that is IR, ex-post IC and weakly budget-balanced under the efficient allocation function. □

5.1. Computational complexity of Algorithm 1

As we already mentioned in this section, any algorithm that checks the existence of a desirable mechanism in the general setting takes exponential time. This is inevitable as the input size is already exponential. Now we analyze the complexity of Algorithm 1 in detail.

Given any efficient allocation, Algorithm 1 can actually be used to determine if there is a payment function that makes the whole mechanism desirable. However, we still need to enumerate the allocation space to find an efficient allocation. Since the total valid data collected from each agent is at most $D \cdot n$, there are $\frac{D\cdot n}{\epsilon} + 1$ possible models that can be allocated to each agent. Therefore, there are at most $\left(\frac{D\cdot n}{\epsilon} + 1\right)^{(D/\epsilon)^n}$ efficient allocations which may tie in terms of social welfare. For each allocation, we analyze the running time of Algorithm 1. The running time of the preprocessing stage is dominated by the preprocessing of computing $Gap_i(t_i', t_i, t_{-i})$, which is $O\left(n \cdot \left(\frac{D}{\epsilon}\right)^{n+1}\right)$. The graph constructed for the Single-Source Shortest-Path algorithm contains at most $O\left(\frac{D^2}{\epsilon^2}\right)$ edges and $O(D/\epsilon)$ vertices, thus the total running time after preprocessing is $O\left(n \cdot (D/\epsilon)^{n-1} \cdot (D/\epsilon)^2 \cdot \log(D/\epsilon)\right) = O\left(n \cdot (D/\epsilon)^{n+1} \cdot \log(D/\epsilon)\right)$. Taking allocation enumeration into account, the complete running time is then $O\left(\left(\frac{D\cdot n}{\epsilon} + 1\right)^{(D/\epsilon)^n} \cdot n \cdot (D/\epsilon)^{n+1} \cdot \log(D/\epsilon)\right)$, which is a double exponential function on n.

Such a running time clearly makes the problem intractable. However, we can circumvent the double-exponent dependence on n by introducing a natural assumption on the market as follows.

Assumption 4 *(Monotone Market). A market is monotone, if the valuation functions $\{v_i\}_{i=1}^n$ satisfies the following constraint:*

$$\sum_{i=1}^{n} v_i(q_i, q_{-i}) < \sum_{i=1}^{n} v_i(q_i', q_{-i}'), \forall q < q',$$

where $q < q'$ means $\forall i \in [n]$, $q_i \le q_i'$ and $\exists\, i \in [n]$, $q_i < q_i'$.

Intuitively, the above assumption says that when an agent adopts a model of a higher quality in the market, the overall market size strictly increases. In essence, this assumption mirrors the idea that as technology advances, the demand for a particular service type expands, drawing in more users. Given such assumption, we can significantly reduce the runtime of searching for a desirable algorithm by only focusing on a unique efficient allocation function.

Proposition 1. *In a monotone market, the only efficient allocation is to assign the best model to every agent.*

Proof. Consider the case when all the agents report truthfully. For each agent, using a better model in the market will strictly increase the social welfare. Since the best model an agent can use is the model trained with all the data from all agents, assigning the best model to every agent is the only efficient allocation. □

The above proposition shows that in a monotone market, there exists a unique efficient allocation. Determining this allocation for each type t takes $O((D/\epsilon)^n)$ time. Therefore, the algorithm's overall running time is $O\left(n \cdot (D/\epsilon)^{n+1} \cdot \log(D/\epsilon)\right)$, which is reduced to a single exponential dependence on n.

6. Conclusion

In this paper, we study the mechanism design problem for multi-party machine learning. We restrict the action space of each agent where he can only misreport a lower type than his actual type and consider the valuation function that is about the allocation outcome and the true types of all agents.

Our results show that in our setting, it is possible to find mechanisms that are simultaneously IR, ex-post IC, efficient and weakly budget-balanced. In the separable valuation setting, we prove that simply charging each agent the profit he gets from the mechanism (i.e., the MEP mechanism) is truthful and revenue-optimal. We also analyze the general setting in a systematical way and provide sufficient and necessary conditions for designing a truthful mechanism. These conditions restrict both the allocation function and the payment function. Finally, we provide an algorithm to find desirable mechanisms that are truthful, individually rational, efficient and weakly budget-balanced simultaneously. Both our sufficient and necessary conditions and algorithmic characterizations indicate that in our setting with a weaker truthfulness constraint (i.e., a larger design space), it is possible to achieve more properties but much harder to find a mechanism satisfying all of them.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Weiran Shen reports financial support was provided by National Natural Science Foundation of China (No. 62106273 & No. 72192805).

References

[1] Y. Hu, D. Niu, J. Yang, S. Zhou, Fdml: a collaborative machine learning framework for distributed features, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2019, pp. 2232–2240.
[2] R. Shokri, V. Shmatikov, Privacy-preserving deep learning, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, 2015, pp. 1310–1321.
[3] V. Smith, C.-K. Chiang, M. Sanjabi, A.S. Talwalkar, Federated multi-task learning, in: Advances in Neural Information Processing Systems, 2017, pp. 4424–4434.
[4] M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 308–318.
[5] R. Yonetani, V. Naresh Boddeti, K.M. Kitani, Y. Sato, Privacy-preserving visual learning using doubly permuted homomorphic encryption, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2040–2050.
[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, 2017, pp. 1273–1282.
[7] L. Blumrosen, M. Feldman, Implementation with a bounded action space, in: Proceedings of the 7th ACM Conference on Electronic Commerce, ACM, 2006, pp. 62–71.
[8] L.M. Ausubel, An efficient ascending-bid auction for multiple objects, Am. Econ. Rev. 94 (2004) 1452–1475.
[9] E. David, A. Rogers, N.R. Jennings, J. Schiff, S. Kraus, M.H. Rothkopf, Optimal design of English auctions with discrete bid levels, ACM Trans. Internet Technol. 7 (2007) 12.
[10] M.S.-Y. Chwe, The discrete bid first auction, Econ. Lett. 31 (1989) 303–306.
[11] A.V. Goldberg, J.D. Hartline, A. Wright, Competitive auctions and digital goods, in: Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2001, pp. 735–744.
[12] A.V. Goldberg, J.D. Hartline, Envy-free auctions for digital goods, in: Proceedings of the 4th ACM Conference on Electronic Commerce, ACM, 2003, pp. 29–35.
[13] W. Fang, P. Tang, S. Zuo, Digital good exchange, in: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 1277–1278.
[14] I. Redko, C. Laclau, On fair cost sharing games in machine learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 4790–4797.
[15] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, D.I. Kim, Incentive design for efficient federated learning in mobile networks: a contract theory approach, arXiv preprint, arXiv:1905.07479, 2019.
[16] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, S. Guo, A survey of incentive mechanism design for federated learning, IEEE Trans. Emerg. Top. Comput. (2021).
[17] W.Y.B. Lim, N.C. Luong, D.T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, C. Miao, Federated learning in mobile edge networks: a comprehensive survey, IEEE Commun. Surv. Tutor. 22 (2020) 2031–2063.

[18] R. Jia, D. Dao, B. Wang, F.A. Hubis, N. Hynes, N.M. Gürel, B. Li, C. Zhang, D. Song, C.J. Spanos, Towards efficient data valuation based on the Shapley value, in: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 1167–1176.

[19] T. Song, Y. Tong, S. Wei, Profit allocation for federated learning, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 2577–2586.

[20] N. Ding, Z. Fang, J. Huang, Optimal contract design for efficient federated learning with multi-dimensional private information, IEEE J. Sel. Areas Commun. 39 (2020) 186–200.

[21] P.R. Milgrom, R.J. Weber, A theory of auctions and competitive bidding, Econometrica 50 (1982) 1089–1122.

[22] T. Roughgarden, I. Talgam-Cohen, Optimal and robust mechanism design with interdependent values, ACM Trans. Econ. Comput. 4 (2016) 18:1–18:34.

[23] S. Chawla, H. Fu, A.R. Karlin, Approximate revenue maximization in interdependent value settings, in: M. Babaioff, V. Conitzer, D.A. Easley (Eds.), ACM Conference on Economics and Computation, EC ’14, Stanford, CA, USA, June 8-12, 2014, ACM, 2014, pp. 277–294.

[24] H. Takabi, E. Hesamifard, M. Ghasemi, Privacy preserving multi-party machine learning with homomorphic encryption, in: NIPS 2016 Workshop on Private Multi-Party Machine Learning, 2016.

[25] C. Mezzetti, Mechanism design with interdependent valuations: efficiency, Econometrica 72 (2004) 1617–1626.

[26] W. Rudin, Functional Analysis, McGraw-Hill, 1991.

[27] H.W. Kuhn, The Hungarian method for the assignment problem, Nav. Res. Logist. Q. 2 (1955) 83–97.

[28] J. Munkres, Algorithms for the assignment and transportation problems, J. Soc. Ind. Appl. Math. 5 (1957) 32–38.

[29] D.P. Bertsekas, A distributed algorithm for the assignment problem, 1979.

[30] D.P. Bertsekas, An auction algorithm for shortest paths, SIAM J. Optim. 1 (1991) 425–447.

[31] D.P. Bertsekas, S. Pallottino, M.G. Scutella, Polynomial auction algorithms for shortest paths, Comput. Optim. Appl. 4 (1995) 99–125.

[32] P.N. Klein, S. Subramanian, A randomized parallel algorithm for single-source shortest paths, J. Algorithms 25 (1997) 205–220.